Supporting Information for In-Focus feature
"YouTube of YouLose" by
Ratmir Derda and Paul J. Bracher.

Overview of the MatLab programs used to process the statistics of the competition.

Websave.m
The script saves the HTML of the 16 web-pages that contained a total of 77 videos and votes and saves them as *.txt files. The name of the file incorporates the time at which the page was saved. On average, each web-page was saved every 10 minutes. The script was run continuously from April 19 to 25, 2011 and at the selected times intervals on April 25-May 6, 2011.

Webparse.m
The script parses the HTML files to identify the regions that contained the number of quotes for each applicant. Script also extracts the time at which the web-page was accessed. The arrays that contain time and vote information, and the unique number of the applicant—the order in which the video-application appeared on the web-page—were saved as a data file votes.mat (for webplot.m and B4andafter.m scripts).

Webplot.m
Plots the results from votes.mat data file as "votes vs. time" plot. The script also calculated the vote differential as [vote(t2) - vote (t1)]/ (t2-t1) and plotted this value as a function of time. The plots were saved as N.jpeg and N.eps, where N is the unique number of the applicant. Some of these plots were used in Figure 3.

B4andafter.m
The script plots the number of votes from votes.mat from the April 25 (before the end of the competition, and before the "illegal votes" were identified by GCC) and votes.mat from May 6 (after the end of the competition and cancellation of the "illegal votes"). To yield the figure 2, the "off-scale" bars (>4500 votes) were tagged manually using Adobe Illustrator.

Note: The text of the scrips is provided for the reference only.
Websave.m, although functional, will not work as written, because the web-page is no longer available. Webparse.m program requires saved HTML pages to run. The samples of the pages are available upon request.

```
****** Websave.m **************

webcore = 'http://gcc.eyeptv.net/submission-list/page/';

for cycling=1:10000
   time1 = clock;
   for i=1:16
      time = clock;
      counter = num2str(i);
      if size(counter) == 1
         counter = ['0' counter];
      end
      webcomplete = [ webcore num2str(i) '/' ];
      filename = [ 'page-' counter '-' num2str(time(3)) '-' num2str(time(4)) '-' num2str(time(5)) '.txt'];
      urlwrite(webcomplete,filename);
      pause_sec = 20*rand;
      pause(pause_sec);
      disp(['cycle=' num2str(cycling) ' N=' counter ' URL=' webcomplete ' file=' filename]);
   end

   time2 = clock;
   diff = time2(5)-time1(5);
   pause_time = 600 - diff*60;
   pause(pause_time);

end %***** EOF **************
```

****** Webparse.m***************

```
dir1= ‘INSER THE COMPLETE PATH TO DIRECTORY THAT CONTAINS SAVED WEB PAGES'
%files = dir(fullfile(dir1, 'page-16-19-15-59.txt'));
files = dir(fullfile(dir1, '*.txt'));
search_st = 'onkeypress="rate_post();" style="cursor: pointer; border: 0px;" /> (<strong>+'
search_end = '</strong> rating/cote, <strong>';
L = size(search_st,2);
vote = [];
votes = cell(80,1);

for k=1:length(files)
   filename=files(k).name;
   pageNum = str2num (filename(6:7));

   %%%% extract time in absolute minutes
   pageTime = filename(9:end-4);
   M = find(pageTime=='-');
   abstime = str2num(pageTime(1:M(1)-1))*24*60 + ...
        str2num(pageTime(M(1)+1:M(2)-1))*60 + ...
        str2num(pageTime(M(2)+1:end));

   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   %%%% monitor the progress
   disp([filename ' .. ' num2str(pageNum) ' .. ' num2str(abstime)]);

   fid=fopen([dir1 '\' filename]);
   counter=0;
   while ~feof(fid)

      tline = fgetl(fid);

      if ~isempty(strfind(tline,search_st))
         d_start = min( strfind(tline,search_st)) + L;
         d_end   = min( strfind(tline,search_end))- 1;
         counter = counter + 1;
         vote = str2num( tline(d_start:d_end));
         contestant = (pageNum-1)*5 + counter;
         votes{contestant} = [votes{contestant}; abstime vote];
         %disp(['contestant=' num2str(contestant) '  vote=' num2str(vote) ]);
      end

   end

   fclose(fid);
end

save([dir1 'votes.mat'],'votes'); %***** EOF **************
```

```
****** Webplot.m ***************
load('votes.mat');
votes2 = cell(80,1);

% sort the votes
for i=1:size(votes,1)
   if size(votes{i},1) == 0
      break;
   end
[temp, IX] = sort ( votes{i}(:,1) );
 temp2 = votes{i}(IX,2);
 votes2{i} = [temp, temp2];
 end

% calculate the vote differential
for i=1:size(votes2,1)
   if size(votes2{i},1) == 0
      break;
   end
  votes2{i}(:,3) = votes2{i}(:,1)/(60*24);
   for ii=2:size(votes{i},1)
     votes2{i}(ii-1,4) = 10*(votes2{i}(ii,2) - votes2{i}(ii-1,2)) / (votes2{i}(ii,1) - votes2{i}(ii-1,1));
   end
  votes2{i}(1,4) = votes2{i}(2,4);
end

% plot the votes and differential, save each figure
for i=1:77

   figure(100);
   set(100,'name',num2str(i), 'position', [100 400 500 150] );
   x = size(votes2{i}(:,4),1);

   % find the axis limits for each set of votes
   if max(votes2{i}(:,4))~=0
      der_lim = 1.2*max(votes2{i}(:,4));
      marg = 0.1* (max(votes2{i}(:,2)) - min(votes2{i}(:,2)));
      vot_lim = [ min(votes2{i}(:,2)) - marg; max(votes2{i}(:,2)) + marg];
   else
      der_lim = 1;
      vot_lim = [ 0.8*min(votes2{i}(:,2)) 1.2*max(votes2{i}(:,2))];
   end

   subplot(1,2,1)
   h2 = plot(votes2{i}(:,3), votes2{i}(:,4));
   set(h2, 'LineStyle', '-', 'color', [0 0 0],...
      'Marker', 'd','MarkerSize', 1, 'MarkerEdgeColor', [0 0 0],'MarkerFaceColor', [0 0 0]);
   set(gca,'outerposition', 'TightInset','FontSize', 7,'ylim', [0, der_lim ]);

   subplot(1,2,2)
    h3 = plot(votes2{i}(:,3), votes2{i}(:,2));
    set(h3, 'LineStyle', '-', 'color', [0 0 0],...
      'Marker', 'd','MarkerSize', 1, 'MarkerEdgeColor', [0 0 0],'MarkerFaceColor', [0 0 0]);
   set(gca,'Outerposition', [0 0 0.45 1/10],'FontSize', 7,'ylim', [vot_lim(1), vot_lim(2) ]);

   counter = num2str(i);
         if size(counter) == 1
            counter = ['0' counter];
         end
   print(100,'-depsc2',[counter '-plot.eps']);
   print(100,'-djpeg ',[counter '-plot.jpg']);
   waitforbuttonpress;
end
```

```
hold off;  %***** EOF **************

****** B4andafter.m ***************

dir1=' PATH TO DIRECTORY THAT CONTAINS PAGES PRIOR TO APRIL 29'
cd(dir1)

load('votes.mat');
for i=1:size(votes)
   if size(votes{i}) == 0
      break
   end
end

num_entries=i-1;
maxv = zeros(num_entries,1);
maxt = zeros(num_entries,1);

for i=1:num_entries
   [maxt(i), I] = max(votes{i}(:,1));
   maxv(i) = votes{i}(I,2);
end

clear votes

dir2= ‘ PATH TO DIRECTORY THAT CONTAINS PAGES AFTER APRIL 29 ’;
cd(dir2)

load('votes.mat');
for i=1:size(votes)
   if size(votes{i}) == 0
      break
   end
end

num_entries=i-1;

maxv2 = zeros(num_entries,1);
maxt2 = zeros(num_entries,1);

for i=1:num_entries
   [maxt2(i), I] = max(votes{i}(:,1));
   maxv2(i) = votes{i}(I,2);
end

[smaxv2, Ind] = sort(maxv2);
 smaxv = maxv(Ind);
bar([smaxv smaxv2], 'grouped');
set(gca,'Yscale','linear', 'Ylim', [10, 4500]);

figure(200)
bar(smaxv2./smaxv);
set(gca,'Yscale','linear', 'Ylim', [0, 1.5]); %***** EOF **************
```